

```

UUU      UUU  EEEEEEEEEEEEEEE  TTTTTTTTTT TTTT  PPPPPPPPPPP  SSSSSSSSSSS  YYY      YYY
UUU      UUU  EEEEEEEEEEEEEEE  TTTTTTTTTT TTTT  PPPPPPPPPPP  SSSSSSSSSSS  YYY      YYY
UUU      UUU  EEEEEEEEEEEEEEE  TTTTTTTTTT TTTT  PPPPPPPPPPP  SSSSSSSSSSS  YYY      YYY
UUU      UUU  EEE              TTT      PPP      PPP  SSS      YYY      YYY
UUU      UUU  EEE              TTT      PPP      PPP  SSS      YYY      YYY
UUU      UUU  EEE              TTT      PPP      PPP  SSS      YYY      YYY
UUU      UUU  EEE              TTT      PPP      PPP  SSS      YYY      YYY
UUU      UUU  EEE              TTT      PPP      PPP  SSS      YYY      YYY
UUU      UUU  EEE              TTT      PPP      PPP  SSS      YYY      YYY
UUU      UUU  EEE              TTT      PPP      PPP  SSS      YYY      YYY
UUU      UUU  EEE              TTT      PPP      PPP  SSS      YYY      YYY
UUU      UUU  EEEEEEEEEEEEEEE  TTT      PPPPPPPPPPP  SSSSSSSSS  YYY
UUU      UUU  EEEEEEEEEEEEEEE  TTT      PPPPPPPPPPP  SSSSSSSSS  YYY
UUU      UUU  EEEEEEEEEEEEEEE  TTT      PPPPPPPPPPP  SSSSSSSSS  YYY
UUU      UUU  EEE              TTT      PPP      SSS      YYY
UUU      UUU  EEE              TTT      PPP      SSS      YYY
UUU      UUU  EEE              TTT      PPP      SSS      YYY
UUU      UUU  EEE              TTT      PPP      SSS      YYY
UUU      UUU  EEE              TTT      PPP      SSS      YYY
UUU      UUU  EEE              TTT      PPP      SSS      YYY
UUUUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE  TTT      PPP      SSSSSSSSSSS  YYY
UUUUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE  TTT      PPP      SSSSSSSSSSS  YYY
UUUUUUUUUUUUUUUUUU  EEEEEEEEEEEEEEE  TTT      PPP      SSSSSSSSSSS  YYY

```

[illegible]

```

LL               IIIIII               SSSSSSSS
LL               IIIIII               SSSSSSSS
LL               II                   SS
LL               II                   SS
LL               II                   SS
LL               II                   SS
LL               II                   SSSSSS
LL               II                   SSSSSS
LL               II                   SS
LL               II                   SS
LL               II                   SS
LL               II                   SS
LLLLLLLLLLLLLL  IIIIII               SSSSSSSS
LLLLLLLLLLLLLL  IIIIII               SSSSSSSS

```

(1)	55	DECLARATIONS
(1)	111	CONDITION TABLES
(1)	148	TM SETUP, TM CLEANUP
(1)	247	CONDITION SUBROUTINES - SETUP AND CLEANUP
(1)	335	FORM CONDS
(1)	428	VERIFY
(1)	569	VFY_CLEANUP

```
0000 1 .TITLE SATSSS42 SATS SYSTEM SERVICE TESTS $FORCEX (SUCC S.C.)
0000 2 .IDENT 'V04-000'
0000 3
0000 4
0000 5 *****
0000 6
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *****
0000 26
0000 27
0000 28
0000 29 ++
0000 30 FACILITY: SYSTST (SATS SYSTEM SERVICE TESTS)
0000 31
0000 32 ABSTRACT:
0000 33
0000 34 THIS MODULE CONTAINS SUBROUTINES WHICH, WHEN LINKED
0000 35 WITH SUCCOMMON.OBJ, FORM TEST MODULE SATSSS42 TO TEST SUCCESSFUL
0000 36 OPERATION OF THE $FORCEX SYSTEM SERVICE. THE SERVICE IS INVOKED
0000 37 UNDER VARIOUS INPUT CONDITIONS WITH VARYING INPUT PARAMETERS. ONLY
0000 38 SUCCESSFUL STATUS CODES ARE EXPECTED IN THIS TEST MODULE. CORRECT
0000 39 OPERATION OF THE SERVICE FOR EACH OF ITS ISSUANCES IS VERIFIED BY
0000 40 CHECKING FOR AN SS$ NORMAL STATUS CODE, EXPECTED RETURN ARGUMENTS
0000 41 AND EXPECTED FUNCTIONALITY PERFORMED.
0000 42
0000 43 ENVIRONMENT: USER MODE IMAGE; NEEDS CMKRNL PRIVILEGE,
0000 44 DYNAMICALLY ACQUIRES OTHER PRIVILEGES, AS NEEDED.
0000 45
0000 46 AUTHOR: THOMAS L. CAFARELLA, CREATION DATE: MAR, 1978
0000 47
0000 48 MODIFIED BY:
0000 49
0000 50 V03-001 LDJ0001 Larry D. Jones, 23-Jun-1983
0000 51 Removed the quota list to force the use of the
0000 52 default sysboot quota values.
0000 53 --
```

```

0000 55      .SBTTL DECLARATIONS
0000 56      :
0000 57      : INCLUDE FILES:
0000 58      :
0000 59      $PRVDEF      ; PRIVILEGE BIT DEFINITIONS
0000 60      $PHDDEF      ; PROCESS HEADER OFFSETS
0000 61      $PQLDEF      ; PROCESS QUOTA CODES
0000 62      $PCBDEF      ; PCB LABELS
0000 63      $LOGDEF      ; LOGICAL NAME TABLE TYPE SYMBOLS
0000 64      $DIBDEF      ; DEVICE INFO BLOCK OFFSETS
0000 65      :
0000 66      : MACROS:
0000 67      :
0000 68      :
0000 69      : EQUATED SYMBOLS:
0000 70      :
0000 71      :
0000 72      : OWN STORAGE:
0000 73      :

```

SATS SYSTEM SERVICE TESTS \$FORCEX (SUCC 16-SEP-1984 00:53:49 VAX/VMS Macro V04-00 Page 3
DECLARATIONS 5-SEP-1984 04:31:22 [UETPSY.SRC]SATSSS42.MAR;1 (1)

```

00000000 75 .PSECT RODATA,RD,NOWRT,NOEXE, LONG
0000 76 TEST_MOD_NAME:: STRING C,<SATSS$42> : TEST MODULE NAME
0009 77 TEST_MOD_NAME_D: STRING I,<SATSS$42> : TEST MODULE NAME DESCRIPTOR
0019 78 MSG1_INP_CTL: STRING I,< SSFEX!4ZW: CONDITIONS:>
0039 79 : FAO CTL STRING FOR MSG1 IN SUCCOMMON.MAR
0039 80 MSG3_ERR_CTL:: STRING I,< *SSFEX!4ZW: !AS>
0051 81 : FAO CTL STRING FOR MSG3 IN SUCCOMMON.MAR
FFFFF7FF DC3CBA00 0051 82 ONE_MIN: .LONG -10*1000*1000*60,-1 : ONE MINUTE (WAKE-UP DELTA)
0059 83 CREATED_PRN: STRING I,<SATSS$42 CRE> : PROCESS & MBX NAME FOR CREATED PROCESS
006D 84 IMAGNAM: STRING I,<SYSTST$RES: SATSUT13.EXE> : IMAGE NAME FOR CREATED PROC
008C 85 LOGNAM_PID: STRING I,<SYSTST$PID> : LOG NAME OF CREATING PID
009E 86 EQUIV_PID: .LONG 4 : EQUIV NAME STRING DESCRIPTOR
00A2 87 .ADDRESS CREATING_PID : ... OF CREATING PID
00A6 88 ;QUOTALIST: $QUOTA CPULM,0 : INFINITE CPU
00A6 89 : $QUOTA BYTLM,512 : BYTE LIMIT FOR BUFFERED I/O
00A6 90 : $QUOTA FILLM,2 : OPEN FILE COUNT LIMIT
00A6 91 : $QUOTA PGFLQUOTA,10 : PAGING FILE QUOTA
00A6 92 : $QUOTA PRCLM,2 : SUBPROCESS QUOTA
00A6 93 : $QUOTA TQELM,3 : TIMER QUEUE ENTRY QUOTA
00A6 94 : $QUOTA LISTEND : DEFINES END OF LIST

```

[illegible]

00000000	96	.PSECT	RWDATA, RD, WRT, NOEXE, LONG	
00000008	97	PRIVMASK:	.BLKQ 1	: ADDR OF PRIVILEGE MASK (IN PHD)
0000000C	98	MBXCHAN:	.BLKL 1	: CHAN. NO. FOR MAILBOX FOR CREATED PROCESS
	99	MBXCHANINFO:		: CHANNEL INFO RETURNED BY GETCHN
00000074	100		.LONG DIB\$K_LENGTH	
00000014	101		.ADDRESS +4	
00000088	102		.BLKB DIB\$K_LENGTH	
0000008C	103	MBXUNIT:	.BLKL 1	: SAVE AREA FOR MAILBOX UNIT NUMBER
	104	MBXBUFF:	STRING 0,120	: MAILBOX BUFFER FOR CREATED PROCESS
00000110	105	DEST_PIDADR:	.BLKL 1	: DESTINATION PID ADDR, WRITTEN BY S.S.
00000114	106	ZEROPID:	.BLKL 1	: PID OF ZEROES
00000000	107	CREATING_PID:	.LONG 0	: PID OF CREATING PROCESS
0000011C	108	CREATED_PID:	.BLKL 1	: PID OF CREATED PROCESS
00000120	109	EXP_STATUS:	.BLKL 1	: EXPECTED STATUS CODE FROM CREATED PROC

SAT
Sym

SYS
SYS
SYS
SYS
SYS
TES
TES
TES
TES
TMD
TM-
TM-
VER
VER
VFY
WOR
WRI
ZER

PSE

\$AB
ROD
RWD
SAT

Pha

Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
515
The
624
51

```
.SBTTL CONDITION TABLES
***** CONDITION TABLES FOR FORCEX SYSTEM SERVICE *****
COND 1,NOTARG,<PID ADDRESS>,-
      <NOT SPECIFIED>,-
      <SPECIFIED, NON-ZERO>,-
      <SPECIFIED, ZERO>,-
      .ADDRESS 0
      .ADDRESS CREATED_PID
      .ADDRESS ZEROPID
COND 2,NOTARG,<PROCESS NAME ADDRESS>,-
      <SPECIFIED>,-
      <NOT SPECIFIED>,-
      .ADDRESS CREATED_PRN
      .ADDRESS 0
COND 3,NOTARG,<CREATED PROCESS TYPE>,-
      <SUBPROCESS>,-
      <DETACHED, DIFFERENT GROUP>,-
      <DETACHED, SAME GROUP, SAME MEMBER>,-
      <DETACHED, SAME GROUP, DIFFERENT MEMBER>,-
      .LONG 0 : PSEUDO-UIC
      .BLKL 1 : UIC
      .BLKL 1 : UIC
      .BLKL 1 : UIC
COND 4,NULL
COND 5,NULL
.PSECT SATSSS42,RD,WRT,EXE
```

00000000	0120	111
	0120	112
	0120	113
	0120	114
	0120	115
	0120	116
	0120	117
	0120	118
	0120	119
00000000	016B	120
00000118	016F	121
00000110	0173	122
	0177	123
	0177	124
	0177	125
	0177	126
	0177	127
00000059	01AD	128
00000000	01B1	129
	01B5	130
	01B5	131
	01B5	132
	01B5	133
	01B5	134
	01B5	135
	01B5	136
00000000	0249	137
00000251	024D	138
00000255	0251	139
00000259	0255	140
	0259	141
	0259	142
	025A	143
	025A	144
	025B	145
00000000		146

```
0000 148 .SBTTL TM_SETUP, TM_CLEANUP
0000 149 :++
0000 150 : FUNCTIONAL DESCRIPTION:
0000 151 :
0000 152 :         TM SETUP AND TM CLEANUP ARE CALLED TO PERFORM
0000 153 : REQUIRED HOUSEKEEPING AT THE BEGINNING AND END, RESPECTIVELY, OF
0000 154 : TEST MODULE EXECUTION.
0000 155 :
0000 156 : CALLING SEQUENCE:
0000 157 :
0000 158 :         BSBW TM_SETUP    BSBW TM_CLEANUP
0000 159 :
0000 160 : INPUT PARAMETERS:
0000 161 :
0000 162 :         NONE
0000 163 :
0000 164 : IMPLICIT INPUTS:
0000 165 :
0000 166 :         NONE
0000 167 :
0000 168 : OUTPUT PARAMETERS:
0000 169 :
0000 170 :         NONE
0000 171 :
0000 172 : IMPLICIT OUTPUTS:
0000 173 :
0000 174 :         TM_SETUP:  COND TABLE INDEX REGISTERS (R2,3,4,5,6) CLEARED;
0000 175 :                   ALL PRIVILEGES ACQUIRED.
0000 176 :
0000 177 : COMPLETION CODES:
0000 178 :
0000 179 :         EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0000 180 :
0000 181 : SIDE EFFECTS:
0000 182 :
0000 183 :         SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0000 184 : (VIA RSB) IF ERROR ENCOUNTERED.
0000 185 :
0000 186 :--
0000 187 :
0000 188 :
0000 189 :
0000 190 TM_SETUP::
0000 191 CLRL R2 ; INITIALIZE
0000 192 CLRL R3 ; .. CONDITION
0000 193 CLRL R4 ; .... TABLE
0000 194 CLRL R5 ; ..... INDEX
0000 195 CLRL R6 ; ..... REGISTERS
0000 196 BSBW MOD MSG PRINT ; PRINT TEST MODULE BEGIN MSG
0000 197 MOVAL TEST_MOD_SUCC,TMD_ADDR ; ASSUME END MSG WILL SHOW SUCCESS
0000 198 INSV #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
0000 199
0000 200 MODE TO,5$,KRNL ; KERNEL MODE TO ACCESS PHD
0000 201 MOVL @#CTL$GL PHD,R9 ; GET PROCESS HEADER ADDRESS
0000 202 MOVAL PHD$Q PRIVMSK(R9),PRIVMASK ; GET PRIV MASK ADDRESS
0000 203 MODE FROM,5$ ; BACK TO USER MODE
0000 204 PRIV ADD,ALL ; GET ALL PRIVILEGES
```

```
52 D4 0000
53 D4 0002
54 D4 0004
55 D4 0006
56 D4 0008
FFF3' 30 000A
00000000'EF 00000000'EF DE 000D
03 00 00000000'8F FO 0018
00000000'EF 0020
59 00000000'9F DO 0048
00000000'EF 69 DE 004F
0056
0057
```

```
0077 204 $SETPRN S TEST MOD_NAME_D ; SET PROCESS NAME
0084 205 SS_CHECK NORMAL ; CHECK STATUS CODE RETURNED FROM SETPRN
00B2 206 $WAKE S PIDADR=CREATING_PID ; GET MY PID
00C1 207 SS_CHECK NORMAL ; CHECK FOR NORMAL RETURN
00EF 208 $HIBER S ; UNDO ABOVE WAKE
00F6 209 SS_CHECK NORMAL ; CHECK FOR NORMAL RETURN
0124 210 $CRELOG_S TBLFLG=#LOG$C_SYSTEM, - ; GET MY PID INTO LOG NAME TABLE
0124 211 LOGNAM=LOGNAM_PID, - ; ... FOR USE BY CREATED PROCESS
0124 212 EQLNAM=EQUIV_PID
2E 50 E8 013B 213 BLBS RO,10$ ; IF SUCCESSFUL, CONTINUE
013E 214 SS_CHECK NORMAL ; USE SS_CHECK TO TERMINATE MODULE
016C 215 10$:
016C 216 :
016C 217 : THE FOLLOWING CODE ESTABLISHES UIC'S IN THE CONDITION 3 TABLE
016C 218 :
016C 219 MODE TO,20$,KRNL ; KERNEL MODE TO ACCESS PCB
59 00000000'9F D0 018F 220 MOVL @#SCH$GL_CURPCB,R9 ; GET CURRENT PCB ADDRESS
59 00BC C9 D0 0196 221 MOVL PCB$L_UIC(R9),R9 ; PICK UP UIC FROM PCB
019B 222 MODE FROM,20$ ; ... AND GET BACK TO USER MODE
019C 223 :
019C 224 : R9 NOW CONTAINS 'MY' UIC
019C 225 :
59 5A 01 9A 019C 226 MOVZBL #1,R10 ; GET COND3 TABLE INDEX NUMBER INTO A REG
00C10000 8F C1 019F 227 ADDL3 #*X10000,R9,COND3_E[R10] ; PUT DIFF GROUP UIC INTO 2ND TABLE ELT
00000249'EF4A 5A D6 01AC 228 INCL R10 ; POINT TO 3RD COND3 TABLE ELEMENT
00000249'EF4A 59 D0 01AE 229 MOVL R9,COND3_E[R10] ; PUT MY UIC INTO TABLE
00000249'EF4A 5A D6 01B6 230 INCL R10 ; POINT TO 4TH COND3 TABLE ELEMENT
00000249'EF4A 59 01 C1 01B8 231 ADDL3 #1,R9,COND3_E[R10] ; PUT DIFF MEMBER UIC INTO THE TABLE
01C1 232 $CREMBX_S CHAN=MBXCHAN, LOGNAM=CREATED PRN, - ; GET MAILBOX FOR PROCESS
01C1 233 MAXMSG=#120, PROMSK=#0, BUFQ00=#240
01E6 234 SS_CHECK NORMAL ; CHECK NORMAL COMPLETION
0214 235 $GETCHN_S CHAN=MBXCHAN, - ; GET CHAN INFO (UNIT NUMBER)
0214 236 PRIBUF=MBXCHANINFO
022E 237 SS_CHECK NORMAL ; CHECK NORMAL COMPLETION
00000088'EF 00000020'EF 3C 025C 238 MOVZWL MBXCHANINFO+8+DIB$W_UNIT,MBXUNIT ; SAVE MAILBOX UNIT NUMBER
05 0267 239 RSB ; RETURN TO MAIN ROUTINE
0268 240 TM_CLEANUP::
0268 241 $DELMBX_S MBXCHAN ; DELETE TERMINATION MAILBOX
0276 242 $DELLOG_S TBLFLG=#LOG$C_SYSTEM, - ; DELETE LOG NAME ACQUIRED ABOVE
0276 243 LOGNAM=LOGNAM_PID
FD76' 30 0287 244 BSBW MOD_MSG_PRINT ; PRINT TEST MODULE END MSG
05 028A 245 RSB ; RETURN TO MAIN ROUTINE
```

```
028B 247 .SBTTL CONDITION SUBROUTINES - SETUP AND CLEANUP
028B 248 :++
028B 249 : FUNCTIONAL DESCRIPTION:
028B 250 :
028B 251 : COND1 AND COND1 CLEANUP ARE SUBROUTINES WHICH ARE EXECUTED
028B 252 : BEFORE AND AFTER THE VERIFY SUBROUTINE, RESPECTIVELY, WHENEVER A NEW
028B 253 : CONDITION X VALUE IS SELECTED (SEE FUNCTIONAL DESCRIPTION OF SUCCOMMON
028B 254 : ROUTINE IN SUCCOMMON.MAR). ANY SETUP FUNCTION PARTICULAR TO THE
028B 255 : CONDITION X TABLE IS INCLUDED IN THE COND1 SUBROUTINE AND CLEANED
028B 256 : UP, IF NECESSARY, IN THE COND1 CLEANUP SUBROUTINE. THIS INCLUDES,
028B 257 : ESPECIALLY, CODE TO DETECT CONFLICTS AMONG CURRENT ENTRIES IN TWO
028B 258 : OR MORE CONDITION TABLES. IF A CONFLICT IS DETECTED, A NON-ZERO
028B 259 : VALUE IS STORED INTO CONFLICT, WHICH CAUSES THE CALLING ROUTINE
028B 260 : (SUCCOMMON) TO SKIP THE CURRENT ENTRY IN THE CONDITION X TABLE.
028B 261 :
028B 262 : CALLING SEQUENCE:
028B 263 :
028B 264 : BSBW COND1 BSBW COND1_CLEANUP
028B 265 : WHERE X = 1,2,3,4,5
028B 266 :
028B 267 : INPUT PARAMETERS:
028B 268 :
028B 269 : CONFLICT = 0
028B 270 :
028B 271 : IMPLICIT INPUTS:
028B 272 :
028B 273 : R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
028B 274 : FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
028B 275 :
028B 276 : OUTPUT PARAMETERS:
028B 277 :
028B 278 : CONFLICT SET TO NON-ZERO IF COND TABLE CONFLICT DETECTED.
028B 279 :
028B 280 : IMPLICIT OUTPUTS:
028B 281 :
028B 282 : R2,3,4,5,6 PRESERVED
028B 283 :
028B 284 : COMPLETION CODES:
028B 285 :
028B 286 : NONE
028B 287 :
028B 288 : SIDE EFFECTS:
028B 289 :
028B 290 : NONE
028B 291 :
028B 292 : --
028B 293 :
028B 294 :
028B 295 :
028B 296 COND1::
05 028B 297 RSB ; RETURN TO MAIN ROUTINE
028C 298 COND1_CLEANUP::
05 028C 299 RSB ; RETURN TO MAIN ROUTINE
028D 300 COND2::
028D 301 CMPL #ZEROPID,COND1_E[R2] ; PID SPECIFIED AS 0 ??
0299 302 BNEQU COND2X ; NO -- NO CONFLICT
029B 303 TSTL COND2_E[R3] ; YES -- IS THERE A PROCESS NAME ??
```

```
0000016B'EF42 00000110'8F
                14
000001AD'EF43
```

```
D1
12
D5
```

```
00000000'EF 00000000'EF 0B 12 02A2 304 BNEQU COND2X ; YES -- NO CONFLICT
; NO -- INDICATE CONFLICT BECAUSE THIS TYPE
; ... OF FORCEX WOULD EXIT CREATING IMAGE
00000000'EF 00000000'EF 90 02A4 305 MOVB ONES,CONFLICT
02AF 306
02AF 307 COND2X:
05 02AF 308 RSB ; RETURN TO MAIN ROUTINE
02B0 309 COND2_CLEANUP::
05 02B0 310 RSB ; RETURN TO MAIN ROUTINE
02B1 311 COND3::
0000016B'EF42 00000118'8F 01 02B1 312 CMPL #CREATED_PID,COND1_E[R2] ; NON-ZERO PID SPECIFIED ?
19 13 02BD 313 BEQLU COND3X ; YES -- NO CONFLICT
000001AD'EF43 05 02BF 314 TSTL COND2_E[R3] ; IS PROCESS NAME SPECIFIED ?
10 13 02C6 315 BEQL COND3X ; NO -- NO CONFLICT
02C8 316
02C8 317 ; NOTE -- AT THIS POINT, PROCESS WILL BE REFERENCED BY PROCESS NAME.
02C8 318
01 54 01 02C8 319 CMPL R4,#1 ; DOES CONDITION 3 SPECIFY DIFFERENT GROUP ?
0B 12 02CB 320 DNEQ COND3X ; NO -- NO CONFLICT
00000000'EF 00000000'EF 90 02CD 321 MOVB ONES,CONFLICT ; YES -- PROCESS NAME FOR DIFF GROUP IS CONF
02D8 322 COND3X:
05 02D8 323 RSB ; RETURN TO MAIN ROUTINE
02D9 324 COND3_CLEANUP::
05 02D9 325 RSB ; RETURN TO MAIN ROUTINE
02DA 326 COND4::
05 02DA 327 RSB ; RETURN TO MAIN ROUTINE
02DB 328 COND4_CLEANUP::
05 02DB 329 RSB ; RETURN TO MAIN ROUTINE
02DC 330 COND5::
05 02DC 331 RSB ; RETURN TO MAIN ROUTINE
02DD 332 COND5_CLEANUP::
05 02DD 333 RSB ; RETURN TO MAIN ROUTINE
```

```
02DE 335      .SBTTL FORM_CONDS
02DE 336      :++
02DE 337      : FUNCTIONAL DESCRIPTION:
02DE 338      :
02DE 339      :           FORM_CONDS FORMATS AND PRINTS INFORMATION ABOUT
02DE 340      : THE CURRENT ELEMENT IN EACH OF THE CONDITION TABLES.
02DE 341      :
02DE 342      : CALLING SEQUENCE:
02DE 343      :
02DE 344      :           BSBW FORM_CONDS
02DE 345      :
02DE 346      : INPUT PARAMETERS:
02DE 347      :
02DE 348      :           NONE
02DE 349      :
02DE 350      : IMPLICIT INPUTS:
02DE 351      :
02DE 352      :           R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
02DE 353      :           FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
02DE 354      :           FOR X = 1,2,3,4,5 :
02DE 355      :               CONDX_T - TITLE TEXT FOR CONDX TABLE
02DE 356      :               CONDX_TAB - ELEMENT TEXT FOR CONDX TABLE
02DE 357      :               CONDX_C - CONTEXT OF THE CONDX TABLE
02DE 358      :               CONDX_E - DATA ELEMENTS OF THE CONDX TABLE
02DE 359      :
02DE 360      : OUTPUT PARAMETERS:
02DE 361      :
02DE 362      :           NONE
02DE 363      :
02DE 364      : IMPLICIT OUTPUTS:
02DE 365      :
02DE 366      :           NONE
02DE 367      :
02DE 368      : COMPLETION CODES:
02DE 369      :
02DE 370      :           NONE
02DE 371      :
02DE 372      : SIDE EFFECTS:
02DE 373      :
02DE 374      :           NONE
02DE 375      :
02DE 376      : --
02DE 377      :
02DE 378      :
02DE 379      :
02DE 380      : FORM_CONDS::
02DE 381      : $FAO_S MSG1_INP_CTL,FAO_LEN,FAO_DESC,TESTNUM
02FD 382      :           : FORMAT CONDITIONS HEADER MSG
02FD 383      : BSBW OUTPUT_MSG           : .. AND PRINT IT
14 00 91 0300 384      : CMPB #COND1_C,#NULL       : IS CONDITION 1 NULL ?
03 12 0303 385      : BNEQU 10$                 : NO -- CONTINUE
00BF 31 0305 386      : BRW FORM_CONDSX          : YES -- SUBROUTINE IS FINISHED
0308 387 10$:
0308 388      : MOVAL COND1_T,MSG_A      : SAVE ADDRESS OF CONDITION 1 TITLE FOR FAO
0313 389      : MOVL COND1_TAB[R2],MSG_B   : SAVE ADDR OF COND 1 CURR TEXT ELT FOR FAO
031F 390      : MOVB #COND1_C,MSG_TXT      : SAVE CONDITION 1 CONTEXT FOR FAO
0326 391      : MOV_VAL COND1_C,COND1_E[R2],MSG_DATA1 : GIVE COND 1 DATA VALUE TO FAO
```

```

      FCD7' 30 0326 392      BSBW WRITE_MSG2      : FORMAT AND WRITE CONDITION 1 MSG
      14 00 91 0329 393      CMPB #COND2_C,#NULL      : IS COND,TION 2 NULL ?
      03 12 032C 394      BNEQU 20$      : NO -- CONTINUE
      0096 31 032E 395      BRW FORM_CONDSX      : YES -- SUBROUTINE IS FINISHED
                                20$:
      00000000'EF 00000177'EF DE 0331 397      MOVAL COND2_T,MSG_A      : SAVE ADDRESS OF CONDITION 2 TITLE FOR FAO
      00000000'EF 0000018D'EF43 DO 033C 398      MOVL COND2_TAB[R3],MSG_B      : SAVE ADDR OF COND 2 CURR TEXT ELT FOR FAO
      00000000'EF 00 90 0348 399      MOVB #COND2_C,MSG_TXT      : SAVE CONDITION 2 CONTEXT FOR FAO
      FCAE' 30 034F 400      MOV VAL COND2_C,COND2_E[R3],MSG_DATA1 : GIVE COND 2 DATA VALUE TO FAO
      14 00 91 0352 401      BSBW WRITE_MSG2      : FORMAT AND WRITE CONDITION 2 MSG
      03 12 0355 402      CMPB #COND3_C,#NULL      : IS CONDITION 3 NULL ?
      006D 31 0357 403      BNEQU 30$      : NO -- CONTINUE
      035A 404      BRW FORM_CONDSX      : YES -- SUBROUTINE IS FINISHED
                                30$:
      00000000'EF 000001B5'EF DE 035A 406      MOVAL COND3_T,MSG_A      : SAVE ADDRESS OF CONDITION 3 TITLE FOR FAO
      00000000'EF 000001CB'EF44 DO 0365 407      MOVL COND3_TAB[R4],MSG_B      : SAVE ADDR OF COND 3 CURR TEXT ELT FOR FAO
      00000000'EF 00 90 0371 408      MOVB #COND3_C,MSG_TXT      : SAVE CONDITION 3 CONTEXT FOR FAO
      FCB5' 30 0378 409      MOV VAL COND3_C,COND3_E[R4],MSG_DATA1 : GIVE COND 3 DATA VALUE TO FAO
      14 14 91 037B 410      BSBW WRITE_MSG2      : FORMAT AND WRITE CONDITION 3 MSG
      47 13 037E 411      CMPB #COND4_C,#NULL      : IS CONDITION 4 NULL ?
      00000000'EF 00000259'EF DE 0380 412      BEQLU FORM_CONDSX      : YES -- SUBROUTINE IS FINISHED
      00000000'EF 00000259'EF45 DO 038B 413      MOVAL COND4_T,MSG_A      : SAVE ADDRESS OF CONDITION 4 TITLE FOR FAO
      00000000'EF 14 90 0397 414      MOVL COND4_TAB[R5],MSG_B      : SAVE ADDR OF COND 4 CURR TEXT ELT FOR FAO
      039E 415      MOVB #COND4_C,MSG_TXT      : SAVE CONDITION 4 CONTEXT FOR FAO
      FCSF' 30 039E 416      MOV VAL COND4_C,COND4_E[R5],MSG_DATA1 : GIVE COND 4 DATA VALUE TO FAO
      14 14 91 03A1 417      BSBW WRITE_MSG2      : FORMAT AND WRITE CONDITION 4 MSG
      21 13 03A4 418      CMPB #COND5_C,#NULL      : IS CONDITION 5 NULL ?
      00000000'EF 0000025A'EF DE 03A6 419      BEQLU FORM_CONDSX      : YES -- SUBROUTINE IS FINISHED
      00000000'EF 0000025A'EF46 DO 03B1 420      MOVAL COND5_T,MSG_A      : SAVE ADDRESS OF CONDITION 5 TITLE FOR FAO
      00000000'EF 14 90 03BD 421      MOVL COND5_TAB[R6],MSG_B      : SAVE ADDR OF COND 5 CURR TEXT ELT FOR FAO
      03C4 422      MOVB #COND5_C,MSG_TXT      : SAVE CONDITION 5 CONTEXT FOR FAO
      FC39' 30 03C4 423      MOV VAL COND5_C,COND5_E[R6],MSG_DATA1 : GIVE COND 5 DATA VALUE TO FAO
      03C7 424      BSBW WRITE_MSG2      : FORMAT AND WRITE CONDITION 5 MSG
      05 03C7 425      FORM_CONDSX:
      03C7 426      RSB      : RETURN TO CALLER
```

```
03CB 428 .SBTTL VERIFY
03CB 429
03CB 430 ++
03CB 431 FUNCTIONAL DESCRIPTION:
03CB 432
03CB 433 VERIFY IS CALLED ONCE FOR EACH COMBINATION OF CONDITION
03CB 434 TABLE VALUES (AS DETERMINED BY THE INDEX REGISTERS R2,3,4,5,6 FOR
03CB 435 COND TABLES 1,2,3,4,5, RESPECTIVELY). VERIFY ESTABLISHES THE CONDITIONS
03CB 436 SPECIFIED BY THE COND TABLES AND ISSUES THE SUBJECT SYSTEM SERVICE
03CB 437 ($FORCEX). THEN, THE SUCCESSFUL OPERATION OF THE SERVICE IS VERIFIED
03CB 438 BY EXAMINING THE STATUS CODE RETURNED, THE VALUES FOR RETURN ARGUMENTS
03CB 439 AND THE FUNCTIONALITY PERFORMED. THE EXAMINATIONS TAKE THE FORM OF
03CB 440 COMPARISONS AGAINST EXPECTED VALUES. ANY FAILING COMPARISON CAUSES AN
03CB 441 ERR_EXIT MACRO TO BE EXECUTED (EITHER DIRECTLY, OR INDIRECTLY,
03CB 442 THROUGH THE SS_CHECK MACRO); ERR_EXIT SETS EFLAG TO NON-ZERO,
03CB 443 PRINTS ERROR MESSAGES AND CAUSES AN IMMEDIATE RSB TO CALLER.
03CB 444 WHEN ERR_EXIT IS EXECUTED, FURTHER CALLS TO VERIFY ARE SUPPRESSED,
03CB 445 AND, AFTER EXECUTING CLEANUP SUBROUTINES, THE IMAGE EXITS.
03CB 446
03CB 447 CALLING SEQUENCE:
03CB 448
03CB 449 BSBW VERIFY
03CB 450
03CB 451 INPUT PARAMETERS:
03CB 452
03CB 453 NONE
03CB 454
03CB 455 IMPLICIT INPUTS:
03CB 456
03CB 457 R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
03CB 458 FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
03CB 459 FOR X = 1,2,3,4,5 :
03CB 460 CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
03CB 461 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
03CB 462 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
03CB 463 FOR CONDX_E.
03CB 464
03CB 465 OUTPUT PARAMETERS:
03CB 466
03CB 467 NONE
03CB 468
03CB 469 IMPLICIT OUTPUTS:
03CB 470
03CB 471 VERIFY HAS NO OUTPUT. SINCE ITS PURPOSE IS TO TEST FOR ERRORS,
03CB 472 IT MERELY RETURNS TO CALLER NORMALLY AFTER THE TESTS, PROVIDING
03CB 473 ALL WERE SUCCESSFUL; IF AN ERROR IS DISCOVERED, RETURN IS VIA
03CB 474 AN ERR_EXIT OR SS_CHECK MACRO, BOTH OF WHICH DOCUMENT DETECTED
03CB 475 ERRORS.
03CB 476
03CB 477 COMPLETION CODES:
03CB 478
03CB 479 EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
03CB 480
03CB 481 SIDE EFFECTS:
03CB 482
03CB 483 SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
03CB 484 (VIA RSB) IF ERROR ENCOUNTERED.
```

```
03C8 485 :--
03C8 486
03C8 487
03C8 488
03C8 489
00000000'EF 95 03C8 490
03 13 03CE 491
FF0B 30 03D0 492
00000110'EF D4 03D3 493
03D3 494
03D9 495
03D9 496
03D9 497
0410 498
0410 499
043E 500
0451 501
047F 502
0486 503
04B4 504
04BF 505
04ED 506
04ED 507
04ED 508
04ED 509
04ED 510
04ED 511
04ED 512
04ED 513
04ED 514
04ED 515
04ED 516
04ED 517
04ED 518
0000016B'EF42 D5 04ED 519
54 12 04F4 520
000001AD'EF43 D5 04F6 521
4B 12 04FD 522
050E 523
050E 524
050E 525
0000011C'EF 00000118'EF D0 053C 526
010E 31 0547 527
054A 528
054A 529
054A 530
054A 531
0000011C'EF 00000114'EF D0 054A 532
0000010C'EF 0000016B'EF42 D0 0555 533
59 000001AD'EF43 D0 0561 534
0569 535
0569 536
0569 537
0569 538
0569 539
00000000'8F 50 D1 057E 540
61 13 0585 541

VERIFY::
TSTB CFLAG ; SHOULD CONDITIONS BE PRINTED ?
BEQL 5$ ; NO -- CONTINUE
BSBW FORM_CONDS ; YES -- FMT & PRINT ALL CONDS FOR THIS T.C.
5$:
CLRL ZEROPIB ; CLEAR ZERO PID
$CREPRC_S PIDADR=CREATED_PID, PRCNAM=CREATED_PRN, -
UIC=COND3_E[R4], IMAGE=IMAGNAM, -
MBXUNT=MBXUNIT; QUOTA=QUOTALIST
; CREATE A PROCESS TO BE FORCEX'D
; ... AND MAKE SURE IT CREATED OK
SS_CHECK NORMAL ; WAKE SELF IN 1 MIN IF CREATED PROC DOESN'T
$SCHDWK S DAYTIM=ONE_MIN ; CHECK FOR NORMAL RETURN
SS_CHECK NORMAL ; SLEEP UNTIL CREATED PROC IS FULLY CREATED
$HBER S ; EXPECT NORMAL RETURN
SS_CHECK NORMAL ; GET RID OF SCHEDULED WAKE-UP
$CANWAK S ; CHECK FOR NORMAL STATUS RETURN
SS_CHECK NORMAL

: SCHEDULED WAKE-UP WILL ONLY BE EFFECTED IF CREATED PROCESS DOES
: NOT GET FULLY CREATED. IN THIS CASE, THE SUBJECT SYSTEM SERVICE
: BELOW WILL FAIL WITH AN APPROPRIATE ERROR CONDITION.

: THE FOLLOWING CODE LOOKS FOR THE SPECIAL CASE OF NO PID SPECIFIED
: AND NO PROCESS NAME SPECIFIED IN CONDITION TABLES. IF THIS CASE
: IS PRESENT, FORCEX IS NOT ISSUED HERE, BUT, INSTEAD, A $WAKE IS
: ISSUED FOR THE CREATED PROCESS, WHICH, IN TURN, ISSUES A $FORCEX
: TO FORCE ITS OWN EXIT. FOR ALL OTHER CASES, THE CREATED PROCESS
: IS FORCED TO EXIT BY A $FORCEX ISSUED HERE IN THIS PROCESS.
:
TSTL COND1_E[R2] ; IS PIDADR SPECIFIED ??
BNEQU 10$ ; YES -- NO SPECIAL CASE -- CONTINUE
TSTL COND2_E[R3] ; NO -- HOW ABOUT PROCESS NAME ??
BNEQU 10$ ; IT EXISTS -- A NORMAL CASE
$WAKE_S PIDADR=CREATED_PID ; NO PIDADR OR PIDADR SPECIFIED
; WAKE CREATED PROCESS TO FORCE ITSELF
SS_CHECK NORMAL ; CHECK FOR NORMAL STATUS RETURN
MOVL CREATED_PID,EXP_STATUS ; ... SET UP EXPECTED STATUS CODE
BRW 20$ ; ... AND GO WAIT FOR ITS MAIL
10$:
; SET UP TO ISSUE SUBJECT $FORCEX IN THIS PROCESS
;
MOVL CREATING_PID,EXP_STATUS ; SET UP EXPECTED STATUS CODE
MOVL COND1_E[R2],DEST_PIDADR ; GET PID ADDRESS OUT OF TABLE
MOVL COND2_E[R3],R9 ; PRCNAM ADDR INTO REG FOR INDIRECT REF'RNCE
:
***** SYSTEM SERVICE CALL WHICH IS THE SUBJECT OF THIS TEST CASE *****
:
$FORCEX_S PIDADR=DEST_PIDADR, PRCNAM=(R9), -
CODE=EXP_STATUS
CMPL R0,$SS$ _NORMAL ; CODE RECEIVED = CODE EXPECTED ?
BEQLU 15$ ; YES -- CONTINUE
```

```
00000000'EF 00000000'8F D0 0587 542      MOVL  #SS$ NORMAL,EXPV      ; NO -- LOAD UP EXPECTED AND ...
00000000'EF 50 D0 0592 543      MOVL  R0,RCV      ; ... RECEIVED VALUES, THEN EXIT
0599 544      ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM FORCEX>
05E8 545 15$:      TSTL  DEST_PIDADR      ; PID RETURNED BY FORCEX ?
0000010C'EF D5 05E8 546      BEQL  20$      ; NO -- KEEP GOING
0000010C'FF 00000118'EF D1 05EE 547      CMPL  CREATED_PID,@DEST_PIDADR ; YES -- IS IT THE CORRECT ONE ?
0000010C'FF 5B 13 05F0 548      BEQL  20$      ; YES -- CONTINUE
00000000'EF 00000118'EF D0 05FB 549      MOVL  CREATED_PID,EXPV      ; NO --LOAD UP EXPECTED AND
00000000'EF 0000010C'FF D0 05FD 550      MOVL  @DEST_PIDADR,RCV      ; ... RECEIVED VALUES, THEN EXIT
0608 551      ERR_EXIT LONG,<INCORRECT PID RETURNED BY FORCEX>
0613 552 20$:
0658 553      :
0658 554      :
0658 555      :
0658 556      :
0658 557      :
0658 558      :
0681 559      :
0681 560      :
0000011C'EF 00000098'EF D1 06AF 561      SS CHECK NORMAL      ; WAIT FOR CREATED PROCESS TO SEND MAIL
0000011C'EF 69 13 06BA 562      CMPL  MBXBUF+12,EXP_STATUS ; CHECK FOR NORMAL STATUS CODE
00000000'EF 0000011C'EF D0 06BC 563      BEQLU VERIFYX      ; CREATED PROC RETURN EXPECTED EXIT STATUS ?
00000000'EF 00000098'EF D0 06C7 564      MOVL  EXP_STATUS,EXPV      ; YES -- ALL IS OK
06D2 565      MOVL  MBXBUF+12,RCV      ; NO -- LOAD UP EXPECTED AND
0725 566      ERR_EXIT LONG,<INCORRECT EXIT STATUS CODE RETURNED IN MAILBOX>
0725 567 VERIFYX:      RSB      ; RETURN TO CALLER
```

```
0726 569 .SBTTL VFY_CLEANUP
0726 570 :++
0726 571 : FUNCTIONAL DESCRIPTION:
0726 572 :
0726 573 : VFY_CLEANUP EXECUTES SYSTEM SERVICES TO UNDO THE
0726 574 : EFFECT OF THOSE ISSUED IN THE VERIFY SUBROUTINE. VFY_CLEANUP MUST
0726 575 : ASSUME THAT VERIFY MAY NOT HAVE EXECUTED IN ITS ENTIRETY (IF AN
0726 576 : ERROR IS FOUND). ALSO, VFY_CLEANUP MAY ISSUE SS CHECK OR ERR_EXIT
0726 577 : ONLY AFTER PERFORMING ALL OF ITS CLEANUP OPERATIONS; THIS IS REQUIRED
0726 578 : IN THE EVENT THAT VFY_CLEANUP IS CALLED DURING ERROR PROCESSING,
0726 579 : WHEN PERFORMING THE REQUIRED CLEANUP IS MORE IMPORTANT THAN
0726 580 : POSSIBLY DISCOVERING A SECOND ERROR.
0726 581 :
0726 582 : CALLING SEQUENCE:
0726 583 :
0726 584 : BSBW VFY_CLEANUP
0726 585 :
0726 586 : INPUT PARAMETERS:
0726 587 :
0726 588 : NONE
0726 589 :
0726 590 : IMPLICIT INPUTS:
0726 591 :
0726 592 : R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
0726 593 : FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
0726 594 : FOR X = 1,2,3,4,5 :
0726 595 : COND_X - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
0726 596 : TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
0726 597 : ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
0726 598 : FOR CONDX_E.
0726 599 :
0726 600 : OUTPUT PARAMETERS:
0726 601 :
0726 602 : NONE
0726 603 :
0726 604 : IMPLICIT OUTPUTS:
0726 605 :
0726 606 : NONE
0726 607 :
0726 608 : COMPLETION CODES:
0726 609 :
0726 610 : EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0726 611 :
0726 612 : SIDE EFFECTS:
0726 613 :
0726 614 : SS CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0726 615 : (VIA RSB) IF ERROR ENCOUNTERED.
0726 616 :
0726 617 : --
0726 618 :
0726 619 :
0726 620 :
0726 621 VFY_CLEANUP::
0726 622 $DELPRI_S PRNAM=CREATED_PRN ; DELETE CREATED PROCESS (IF STILL HERE)
05 0735 623 RSB ; RETURN TO CALLER
0736 624 .END
```

=	00000010	G	
	0000010C	R	03
=	00000074		
=	0000000C		
	*****	X	04
	0000009E	R	02
	*****	X	04
	0000011C	R	03
	*****	X	04
	*****	X	04
	000002DE	RG	04
	000003C7	R	04
	0000006D	R	02
	*****	X	04
=	00000000		
	0000008C	R	02
=	00000004	G	
	0000008C	R	03
	00000008	R	03
	0000000C	R	03
	00000088	R	03
	*****	X	04
	*****	X	04
	00000019	R	02
	00000039	RG	02
	*****	X	04
	*****	X	04
	*****	X	04
=	00000000	G	
=	00000014	G	
	*****	X	04
	00000051	R	02
	*****	X	04
=	0000008C		
	*****	X	04
=	00000000		
	00000000	R	03
=	00000002		
	*****	X	04
=	00000008	G	
	*****	X	04
	*****	X	04
	*****	X	04
	*****	X	04
	*****	X	04
	*****	X	04
	*****	GX	04
	*****	GX	04
	*****	GX	04
	*****	GX	04
	*****	GX	04
	*****	GX	04
	*****	GX	04
	*****	GX	04
	*****	X	04
	*****	GX	04
	*****	GX	04

SATSSS42
Symbol table

SATS SYSTEM SERVICE TESTS \$FORCEX (SUCC 16-SEP-1984 00:53:49 VAX/VMS Macro V04-00
5-SEP-1984 04:31:22 [UETPSY.SRC]SATSSS42.MAR;1

Page 17
(1)

SYSSHIBER	*****	GX	04
SYSQIOW	*****	GX	04
SYSSCHDWK	*****	GX	04
SYSSSETPRN	*****	GX	04
SYSSSETPRV	*****	GX	04
SYSSWAKE	*****	GX	04
TESTNUM	*****	X	04
TEST_MOD_NAME	00000000	RG	02
TEST_MOD_NAME_D	00000009	R	02
TEST_MOD_SUCC	*****	X	04
TMD_ADDR	*****	X	04
TM_CLEANUP	00000268	RG	04
TM_SETUP	00000000	RG	04
VERIFY	000003C8	RG	04
VERIFYX	00000725	R	04
VFY_CLEANUP	00000726	RG	04
WORD	= 00000002	G	
WRITE_MSG2	*****	X	04
ZEROPID	00000110	R	03

! Psect synopsis !

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
RODATA	000000A6 (166.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
RWDATA	00000258 (603.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SATSSS42	00000736 (1846.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

! Performance indicators !

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.07	00:00:00.37
Command processing	113	00:00:00.61	00:00:02.18
Pass 1	316	00:00:09.72	00:00:20.20
Symbol table sort	0	00:00:00.79	00:00:00.81
Pass 2	136	00:00:02.34	00:00:03.22
Symbol table output	16	00:00:00.10	00:00:00.12
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	614	00:00:13.66	00:00:26.93

The working set limit was 1350 pages.

51508 bytes (101 pages) of virtual memory were used to buffer the intermediate code.

There were 30 pages of symbol table space allocated to hold 510 non-local and 48 local symbols.

624 source lines were read in Pass 1, producing 26 object records in Pass 2.

51 pages of virtual memory were used to define 41 macros.

! Macro library statistics !

Macro library name

Macros defined

_ \$255\$DUA28:[SHRLIB]UETP.MLB;1
- \$255\$DUA28:[SYS.OBJ]LIB.MLB;1
- \$255\$DUA28:[SYS:LIB]STARLET.MLB;2
TOTALS (all libraries)

8
3
27
38

934 GETS were required to define 38 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:SATSSS42/OBJ=OBJ\$:SATSSS42 MSRC\$:SATSSS42/UPDATE=(ENH\$:SATSSS42)+EXECML\$/LIB+SHRLIB\$:UETP/LIB

0423

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY